NASA TECHNICAL MEMORANDUM

NASA TM X-62,496

NASA TM X-62,496

# A TECHNIQUE FOR ACCELERATING ITERATIVE CONVERGENCE IN NUMERICAL INTEGRATION, WITH APPLICATION IN TRANSONIC AERODYNAMICS

E. Dale Martin

NASA—Ames Research Center
Moffett Field, California 94035

October 1975

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| TM X-62,496 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| A TECHNIQUE FOR ACCELERATING ITERATIVE CONVERGENCE IN NUMERICAL INTEGRATION, WITH APPLICATION IN TRANSONIC AERODYNAMICS | 6. Performing Organization Code |

| 7. Author(s) | 8. Performing Organization Report No. |
|---|---|
| E. Dale Martin | A-6311 |
| | 10. Work Unit No. 505-06-12-08-00-21 |
| 9. Performing Organization Name and Address | 11. Contract or Grant No. |
| Ames Research Center<br>Moffett Field, California | 13. Type of Report and Period Covered |
| 12. Sponsoring Agency Name and Address | Technical Memorandum |
| National Aeronautics and Space Administration<br>Washington, D.C. | 14. Sponsoring Agency Code |

15. Supplementary Notes

16. Abstract

A technique is described for the efficient numerical solution of nonlinear partial differential equations by rapid iteration. In particular, a special approach is described for applying the Aitken acceleration formula (a simple Padé approximant) for accelerating the iterative convergence. The method finds the most appropriate successive approximations, which are in a most nearly geometric sequence, for use in the Aitken formula. Simple examples are given to illustrate the use of the method. The method is then applied to the mixed elliptic-hyperbolic problem of steady, inviscid, transonic flow over an airfoil in a subsonic free stream.

| 17. Key Words (Suggested by Author(s)) | | 18. Distribution Statement |
|---|---|---|
| Transonic aerodynamics | Aitken extrapolation | STAR Categories: 02, 64 |
| Numerical analysis | | |
| Iteration | | |
| Convergence acceleration | | |
| Relaxation | | |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price* |
|---|---|---|---|
| Unclassified | Unclassified | 18 | $3.25 |

*For sale by the National Technical Information Service, Springfield, Virginia 22151

# A Technique for Accelerating Iterative Convergence in Numerical Integration, with Application in Transonic Aerodynamics

E. DALE MARTIN
NASA Ames Research Center
Moffett Field, California, USA

**Summary**

A technique is described for the efficient numerical solution of nonlinear partial differential equations by rapid iteration. In particular, a special approach is described for applying the Aitken acceleration formula (a simple Padé approximant) for accelerating the iterative convergence. The method finds the most appropriate successive approximations, which are in a most nearly geometric sequence, for use in the Aitken formula. Simple examples are given to illustrate the use of the method. The method is then applied to the mixed elliptic-hyperbolic problem of steady, inviscid, transonic flow over an airfoil in a subsonic free stream.

## 1. Introduction

The numerical solutions of nonlinear partial differential equations such as those governing fluid flows frequently are obtained most efficiently by iterative methods. The rate of iterative convergence of the method chosen is an important consideration, and various means of accelerating the iterative convergence have been useful.

One popular device for accelerating convergence of a sequence of numbers such as provided by iteration is Aitken's extrapolation formula (or $\Delta^2$ process) [1], whose use is described in most books on numerical methods [2] and which is identified [3–5] as a simple Padé approximant if the successive iterates are partial sums of a power series. Shanks [3] provided generalizations of Aitken's transformation and studied their use. In [6] Wynn gave a simple algorithm for rapid computation of one of the nonlinear transforms studied by Shanks, and later Wynn [7] discussed application of this acceleration technique to vector and matrix problems, including application to boundary-value and initial-value problems.

The present paper describes a special technique for applying the Aitken extrapolation formula for accelerating iterative convergence in the numerical solution of partial differential equations. The method was first introduced and used in [8] and then used in a modified form in [9] with additional results given in [9,10]. Although the application to be discussed is in a numerical finite-difference solution, the general method applies equally well, for example, to analytical solutions or to numerical solutions by finite-element methods. The use of the simple Aitken formula with three successive iterates is emphasized

(even though the elegant $\epsilon$-algorithm of Wynn with longer sequences could be used), because the eventual applications are expected to be those numerical problems requiring significant computer storage. The Aitken formula, using only three iterates, requires less storage than other forms of the $\epsilon$-algorithm.

Often the use of the Aitken formula with iterates obtained arbitrarily by successive approximations does not lead to a significantly improved approximation. However, because Shanks [3] showed that the formula works best if the sequence is "nearly geometric," the present approach seeks to obtain successive iterates that are in a nearly geometric sequence. (Because of the work of Shanks in popularizing the Aitken formula and his valuable demonstration of the special applicability to "nearly geometric sequences," our past work has referred to the simple extrapolation formula as the "Aitken/Shanks formula.") The sequence of approximations can be most nearly geometric if obtained from a power-series construction. Therefore, the basis of the present approach is the construction of successive approximations derived from formal power-series expansions to obtain as closely as possible a nearly geometric sequence. The technique is based on the concepts of perturbation-series expansions (in the sense of Poincaré; see Bellman [11]). An artificial parameter is introduced in such a way as to obtain three problems to solve for terms of a nearly geometric series, for use in the Aitken/Shanks formula. Expansion in powers of an artificial parameter has also been considered by Genz [5] to develop a mathematical proof (unknown by the authors of [8] at that writing), but the central idea in the present approach is that the artificial-parameter expansions are used, in combination with an "artificially extended form" of the equations to be solved, as a device to determine most appropriate successive approximations. This technique produces the nearly geometric sequence of solutions, even in nonlinear problems. The previous application of the Aitken/Shanks transformations to acceleration of iterations in numerical integration by Wynn [7] used simple straightforward iterations. The results of such a procedure with use of only the simplest acceleration formula are described below for an example problem and are compared with the present method.

The present approach based on perturbation series requires that complete perturbation solutions be available on the entire computation field (or entire domain of the equations) at each iteration. This concept therefore adapts well to a finite-difference method using "direct elliptic solvers" [12–15] in the iterative procedure to determine the solution simultaneously at all points on the entire computation field (rather than in successive traverses over the field as in a point- or line-relaxation method). Such methods have been referred to as "semidirect" [8–10].

After several simple examples to illustrate the method, it is applied to the problem of inviscid flow over an airfoil in a subsonic free stream, including conditions for which the flow equations are of mixed type (elliptic in an outer region, with an embedded hyperbolic region and a shock wave). This transonic-aerodynamic-flow problem has also been treated by Hafez and Cheng [16] using the Aitken/Shanks acceleration formula, but in a quite different way, in combination with a line-relaxation method.

## 2. General Formulation of Method

Consider the general partial-differential or difference equation system and the accompanying boundary conditions represented by

$$\mathbf{LU} - \mathbf{F(x)} = \mathbf{NU} \quad \text{in } R, \tag{2.1}$$

$$\mathbf{BU} = \mathbf{G(x)} \quad \text{on } B, \tag{2.2}$$

where $\mathbf{U} = \mathbf{U(x)}$ is a vector function of the position vector $\mathbf{x}$, $\mathbf{L}$ is a separable, linear, elliptic differential or difference operator, $\mathbf{F(x)}$ is a given vector function and $\mathbf{N}$ is a possibly nonlinear operator such that the operation $\mathbf{NU}$ is a vector of the same dimension as $\mathbf{U}$ and has components that may involve $\mathbf{U}$, $\mathbf{x}$, and derivatives of the components of $\mathbf{U}$ with respect to the components of $\mathbf{x}$. Assume for simplicity that $\mathbf{B}$ is a linear operator The boundary condition (2.2) is applied on $B$, which includes all appropriate boundary segments of the domain $R$. For illustration of this notation and of the method, simple one-dimensional examples are given in the next section. Examples treated in the earlier version of [8] included (i) the scalar Laplacian as $\mathbf{L}$ with a scalar, $\psi$, as $\mathbf{U}$, and (ii) a Cauchy-Riemann operator matrix as $\mathbf{L}$ with two components of $\mathbf{U}$, denoted as $u$ and $v$. The right side of (2.1) can be complicated and can make the equation system hyperbolic or parabolic in some regions [8–10].

In the formulation of a problem to be solved, $\mathbf{L}$ and $\mathbf{F(x)}$ are chosen judiciously and may be the result of "scaling and shifting" transformations [17,9] for increasing the rate of iterative convergence or of addition of terms [9,10] for stabilizing iterations. For treatment with additional terms, an extended Cauchy-Riemann solver for use in present calculations has been described in [18].

In the methods to be discussed for the iterative solution of eqs. (2.1) and (2.2), suppose $\mathbf{U}_1\mathbf{(x)}$, $\mathbf{U}_2\mathbf{(x)}$, and $\mathbf{U}_3\mathbf{(x)}$ are successive approximations to $\mathbf{U(x)}$ in $R$. Let $u(x)$ and $u_n(x)$ be respectively each a single scalar component of the vectors $\mathbf{U(x)}$ and $\mathbf{U}_n\mathbf{(x)}$ $(n = 1,2,3)$. Then one form of the Aitken/Shanks extrapolation formula [1,3] for an improved approximation $u^*(x)$ to $u(x)$ is

$$u^*(x) = \frac{u_1u_3 - u_2^2}{u_1 - 2u_2 + u_3} \tag{2.3}$$

Application of the formula in this way to individual components of $\mathbf{U}$ at each $\mathbf{x}$ separately is referred to by Wynn [7] as use of a "primitive inverse" of the $\epsilon$-algorithm. Wynn concludes that use of the primitive inverse is competitive with use of other more complicated inverses. The work of Hafez and Cheng [16] considers coupling of the matrix elements in the numerical solution, which is related to the more complex inverses of the $\epsilon$-algorithm.

**2.1 Artificially extended equation.** For obtaining power-series solutions to (2.1) and (2.2) that are most appropriate for use in the Aitken/Shanks extrapolation formula, it has been found convenient to artificially extend eq. (2.1) by inserting both an artificial parameter $\epsilon$ and an "initial approximation," $\mathbf{U}_0\mathbf{(x)}$, to $\mathbf{U(x)}$ as follows. Let

$$\mathbf{LU} - \mathbf{F(x)} = (1-\epsilon)\mathbf{NU}_0 + \epsilon\,\mathbf{NU} \quad \text{in } R \tag{2.4}$$

along with condition (2.2). Note that the solution $U$ to (2.4) with (2.2) depends on $\epsilon$ (as well as on the specified function $U_0(x)$ ): $U = U(x,\epsilon)$. However, at $\epsilon = 1$, the solution to (2.4) with (2.2) is the same as the solution to the original equations (2.1) with (2.2). Furthermore, if $U_0(x)$ is close to the solution $U(x)$, then (2.4) is nearly the same as (2.1) and the solutions then are nearly the same. Thus, either of the conditions $\epsilon = 1$ or $U_0 = U$ makes (2.4) the same as (2.1). Both of these facts can be used to advantage in the methods to be discussed.

**2.2 Method 1.** The simplest iteration scheme is a straightforward method of successive approximations. Although this method can be combined with use of a relaxation parameter (see [8,9]), for simplicity here we omit that useful device. If we let $\epsilon = 0$ in (2.4) and define $U_0(x)$ as a previous iteration, we obtain the following equations for the iterative solution denoted as Method 1(a):

$$LU_n - F = NU_{n-1} \quad \text{in } R, \tag{2.5a}$$

$$BU_n = G(x) \quad \text{on } B, \tag{2.5b}$$

where subscript $n$ denotes iteration number.

If, as is frequently done, the Aitken/Shanks formula is used to attempt to accelerate the convergence of the iteration, we denote as Method 1(b) the solution of (2.5) for three successive iterates and substitution of the results for one component of each $U_n$ into (2.3). (This designation of Method 1(b) is useful for a comparison in an example problem below.)

**2.3 Method 2.** The new approach for applying the Aitken/Shanks formula, first introduced and used in [8] and in a modified form in [9], is referred to as Method 2. The two versions are called, respectively, Methods 2(a) and 2(b) for later convenience.

Consider the solution to (2.4) with condition (2.2). The solution evaluated at $\epsilon = 1$ is a solution to (2.1) with (2.2). The specified $U_0(x)$ can be used as an initial approximation to $U$. For obtaining a most nearly geometric sequence of approximations, assume that

$$U(x,\epsilon) \sim U_1'(x) + \epsilon U_2'(x) + \epsilon^2 U_3'(x) + \ldots . \tag{2.6}$$

Successive approximations to $U(x)$ are then defined by n-term truncations of the series (2.6):

$$U_n = \sum_{i=1}^{n} \epsilon^{i-1} U_i'(x) \tag{2.7}$$

Although (2.6) is equivalent to a Taylor series or asymptotic series expansion about $\epsilon = 0$, its convergence or lack of convergence at $\epsilon = 1$ is not of particular significance for applicability of eq. (2.3) (see [3]). If the series (2.6) is substituted into the problem of eq. (2.4) and condition (2.2) and coefficients of powers of $\epsilon$ are collected, one obtains equations to solve for the $U_n'$:

$$LU_1' - F = NU_o \quad \text{in } R; \qquad BU_1' = G(x) \quad \text{on } B; \tag{2.8a}$$

$$LU_2' = NU_1' - NU_o \quad \text{in } R; \qquad BU_2' = 0 \quad \text{on } B; \tag{2.8b}$$

$$LU_3' = N_2' \{U_2', U_1'\} \quad \text{in } R; \qquad BU_3' = 0 \quad \text{on } B; \tag{2.8c}$$

in which $N_2'$ is defined by the perturbation expansion

$$NU = NU_1' + \epsilon N_2' \{U_2', U_1'\} + O(\epsilon^2). \tag{2.9}$$

With the definitions (2.7) and

$$N_2 \{U_2, U_1\} \equiv NU_1' + \epsilon N_2' \{U_2', U_1'\} \tag{2.10}$$

one can also solve the following equations for the successive approximations, $U_n$:

$$LU_1 - F = NU_o \quad \text{in } R; \qquad BU_1 = G(x) \text{ on } B \tag{2.11a}$$

$$LU_2 - F = NU_1 \quad \text{in } R; \qquad BU_2 = G(x) \text{ on } B \tag{2.11b}$$

$$LU_3 - F = N_2 \{U_2, U_1\} \quad \text{in } R; \qquad BU_3 = G(x) \text{ on } B \tag{2.11c}$$

in which it has been assumed that $\epsilon = 1$. Note that if the right side of eq. (2.1) is linear in $U(x)$, then the problems for the successive $U_n$ in eqs. (2.11) are the same as (2.5) for Method 1.

We denote as Method 2(a) the solution of eqs. (2.11) for three successive iterates and substitution of the results for one component of each $U_n$ into (2.3) to obtain an improved approximation. (If $NU$ is linear in $U$, this is the same as Method 1(b)). Note that when the solution is near to convergence at any $x$, significant errors will be introduced by the loss of significant figures in applying eq. (2.3).

An alternative procedure (denoted as Method 2(b)) that eliminates the difficulty near convergence is to replace eq. (2.3) by the equivalent expression (at $\epsilon = 1$):

$$u^*(x) = u_1' - \frac{(u_2')^2}{u_3' - u_2'}, \tag{2.12}$$

where each $u_n'(x)$ is a single component of the vector $U_n'(x)$. That is, eqs. (2.8) are solved for $U_n'(x)$, and (2.12) is used for extrapolation.

In a numerical solution, $u^*(x)$ can be used as the next $u_o(x)$ in a repetition of the sequence.

## 3. Example Problems and Comparison of Methods

This section gives simple analytical one-dimensional examples for illustration and comparison of the methods.

**3.1 Example 1.** Consider the nonlinear problem

$$(d/dx + 1)u = (1/2)u^2 \quad \text{in } 0 \leqslant x < \infty, \tag{3.1a}$$

$$u(0) = 1. \tag{3.1b}$$

The iterative solution by Method 1 is found from

$$(d/dx + 1)u_n = (1/2)u_{n-1}^2 \quad , \quad u_n(0) = 1 . \tag{3.2}$$

The analytical solutions for $n = 1,2,3$ (assuming $u_0 = 0$) are:

$$u_1(x) = e^{-x} , \tag{3.3a}$$

$$u_2(x) = e^{-x} [1 + p(x)] , \tag{3.3b}$$

$$u_3(x) = e^{-x} [1 + p(x) + p^2(x) + \tfrac{1}{3} p^3(x)] . \tag{3.3c}$$

where

$$p(x) = (1/2)(1-e^{-x}) . \tag{3.4}$$

For Method 2, the artificially extended equation is:

$$(d/dx + 1)u = (1-\epsilon)(1/2)u_0^2 + \epsilon(1/2)u^2 , \tag{3.5a}$$

$$u(0) = 1 . \tag{3.5b}$$

Substitution of

$$u = u_1'(x) + \epsilon u_2'(x) + \epsilon^2 u_3'(x) + \ldots \tag{3.6}$$

into (3.5) leads to

$$(d/dx + 1)u_1' = (1/2)u_0^2 , \qquad u_1'(0) = 1 , \tag{3.7a}$$

$$(d/dx + 1)u_2' = (1/2)[(u_1')^2 - u_0^2] , \qquad u_2'(0) = 0 , \tag{3.7b}$$

$$(d/dx + 1)u_3' = u_1' u_2' , \qquad u_3'(0) = 0 , \tag{3.7c}$$

or equivalently, with $\epsilon = 1$ and eq. (2.7),

$$(d/dx + 1)u_1 = (1/2)u_0^2 , \qquad u_1(0) = 1 , \tag{3.8a}$$

$$(d/dx + 1)u_2 = (1/2)u_1^2 , \qquad u_2(0) = 1 , \tag{3.8b}$$

$$(d/dx + 1)u_3 = (1/2)u_2^2 - (1/2)(u_1-u_2)^2 , \qquad u_3(0) = 1 . \tag{3.8c}$$

The analytical solutions to (3.7) with $u_0 = 0$ are:

$$u_n'(x) = e^{-x}[p(x)]^{n-1} , \tag{3.9}$$

where $p(x)$ is given by (3.4) and where the solutions $u_n$ to (3.8) are given by (2.7). Evaluations of these solutions at $x = 1$ and applications of the appropriate forms of the Aitken/Shanks formula are given in Table 1. The results for the extrapolated solution $u^*$ may be compared with the exact solution to (3.1),

$$u(x) = 2(1+e^x)^{-1} , \tag{3.10}$$

Table 1. Results of Example 1 at $x = 1$ ($u_0 = 0$)

| METHOD: | 1(b) | 2(a) | 2(b) |
|---|---|---|---|
| EQUATIONS: | (3.2) & (2.3) | (3.8) & (2.3) | (3.7) & (2.12) |
| n | $u_n(1)$ | $u_n(1)$ | $u_n'(1)$ |
| 1 | 0.3678794412 | 0.3678794412 | 0.3678794412 |
| 2 | .4841515202 | .4841515202 | .2325441579 |
| 3 | .5247721376 | .5209005060 | .1469959430 |
| $u^*(1)$ = | .546583145 | .537882842 | .5378828426 |
| Exact $u(1)$ = | .5378828428 | .5378828428 | .5378828428 |

evaluated at $x = 1$: $u(1) = 0.5378828428$ to ten significant figures. We note first that the extrapolated solution $u^*$ by Method 1(b) is somewhat closer to the exact value than $u_3$, but not significantly closer. We note further that the third approximation, $u_3$, by Method 2(a) is not as good an approximation as $u_3$ in Method 1, but that the extrapolated solutions by Methods 2(a) and 2(b) are exact except for loss of 1 or 2 significant figures. (Method 2(a) is less exact because of loss of significant figures in (2.3).) The striking accuracy of Method 2 in this example occurs because the sequence of solutions produced by Method 2 is precisely geometric, i.e. $u_{n+1}'/u_n'$ = constant for all n at a given x. The difference from Method 1 is seen by comparing eqs. (3.2) with (3.8), in which (3.8c) has an additional term that produces the geometric sequence.

3.2 **Example 2.** Consider next an example which is linear (so that Method 2(a) would give the same results as Method 1(b)), but for which the iterative sequence is "nearly geometric." Let us use Method 2(b) for this example (eqs. (2.8) with (2.6), (2.7), and (2.12)).

The problem is

$$\frac{du}{dx} - 2 = -x\frac{du}{dx} - 2u \quad \text{in } 0 \leqslant x < \infty \ , \ u(0) = 0 \ , \tag{3.11}$$

which is written in this way in analogy to more complex problems in which one may put a very simple operator on the left and the rest of the terms on the right for iteration. (One can also shift the term $2u$ to the left side, with very similar results.) The artificially extended equation is

$$\left.\begin{aligned} \frac{du}{dx} - 2 = (1-\epsilon)\left(-x\frac{du_0}{dx} - 2u_0\right) + \epsilon\left(-x\frac{du}{dx} - 2u\right) \quad \text{in } 0 \leqslant x < \infty \\ u(0) = 0 \end{aligned}\right\} \tag{3.12}$$

Substitution of (3.6) leads to (with $u_0 = 0$):

$$du_1'/dx - 2 = 0, \qquad u_1'(0) = 0, \tag{3.13a}$$

$$du_2'/dx = -x\,du_1'/dx - 2u_1', \qquad u_2'(0) = 0, \tag{3.13b}$$

$$du_3'/dx = -x\,du_2'/dx - 2u_2', \qquad u_3'(0) = 0. \tag{3.13c}$$

The analytical solutions are

$$u_n'(x) = (-1)^{n+1}(n+1)x^n \tag{3.14}$$

and the successive approximations are given by (2.7). The sequence (3.14) is not geometric, but since $\lim_{n\to\infty}[u_{n+1}'(x)/u_n'(x)]$ exists at given x, the sequence is "nearly geometric" [3]. Evaluation of the solutions (3.14) at $x = 0.5$ gives $(u_1', u_2', u_3') = (1.00, -.75, .50)$ so that the successive approximations are $(u_1, u_2, u_3) = (1.00, .25, .75)$. Substitution of the $u_n'$ into (2.12) gives $u^*(.5) = 0.55$, which compares well with the exact solution to (3.11),

$$u(x) = (2x + x^2)(1 + x)^{-2}, \tag{3.15}$$

from which $u(0.5) = 5/9 = 0.555555\ldots$.

## 4. Transonic Flow Over an Airfoil

For application of the methods described above, consider two-dimensional, steady, inviscid flow over a thin symmetrical parabolic-arc airfoil in a subsonic free stream. At high subsonic Mach numbers, part of the flow can be supersonic, so we consider the transonic small-disturbance equations, which are nonlinear elliptic partial differential equations in subsonic regions and hyperbolic equations in supersonic regions. Transition of the velocity field from a subsonic region to the embedded supersonic zone is smooth, but transition from the supersonic to subsonic region is usually discontinuous, through a shock wave. The improved finite-difference method of Murman and Cole [19–22] captures the shock waves (in a fully conservative way) but spreads the rapid transition over several mesh points.

In [8] a semidirect finite-difference method, based on the use of a fast direct Cauchy-Riemann solver [15], was applied to solving the equivalent of Murman's transonic finite-difference equations [21] iteratively for the perturbation velocities, u and v. (The iteration procedure has been formulated in such a way that at nonelliptic points terms on the right side of the difference equations cancel out the elliptic character of the left side when the iterated solution converges.) Both Methods 1(a) and 2(a) described above worked well for subcritical and for slightly supercritical (local Mach number > 1) flows, except that Method 2(a) could be used only before any part of the solution was nearly converged. In [9] the method was extended to strongly supercritical flow by the addition of stabilizing terms to the difference equations and to the Cauchy-Riemann solver [18]. Also introduced in [9] was the method version denoted here as Method 2(b), which can be used when the solution is nearly converged. In smooth subsonic flows the acceleration technique is effectively used repeatedly. However, in transonic

flows with strong shock waves, the acceleration technique is not helpful at the beginning of the iteration when the shock wave and its location are not well defined. Therefore in [9] it was considered desirable to use the straightforward iteration Method 1(a) until the maximum residual is reasonably small, so that the supersonic region is nearly defined, and then use Method 2(b) to extrapolate three iterates to a final solution. A fully conservative second-order-accurate formulation has been introduced in [10], and so a fomulation that includes either Murman's fully-conservative first-order-accurate formulation or the second-order formulation will be used here.

**4.1 Governing equations and boundary conditions.** Let the dimensionless X and Y axes be respectively along and normal to the airfoil chord, the free-stream Mach number be $M_\infty < 1$, and the dimensionless velocity components in the X and Y directions be U,V. One may then define perturbation velocity components u.v through a Prandtl-Glauert transformation with $\beta \equiv (1 - M_\infty^2)^{1/2}$:

$$U = 1 + (\tau/\beta)u\,, \quad V = \tau v\,, \quad Y = y/\beta\,, \quad X = x\,, \tag{4.1}$$

which amounts to shifting and scaling of certain terms (cf. [17, 8–10], so that the transonic small disturbance equations take the form

$$f_X + g_Y = 0\,, \quad u_Y - v_X = 0 \tag{4.2a,b}$$

where

$$f = f(u) = u - au^2\,, \quad g = g(v) = v\,, \tag{4.3a,b}$$

$$a = \tau(\gamma+1)M_\infty^2/2\beta^3\,, \tag{4.4}$$

in which a is a transonic similarity parameter and $\tau$ is an airfoil thickness ratio. Eqs. (4.2) are often written in terms of a perturbation velocity potential $\phi$ defined by $u = \phi_X$, $v = \phi_Y$, and all the developments to be described apply as well to that potential equation.

The equation system (4.2) is elliptic, parabolic, or hyperbolic depending on whether $u - u_{CR}$ is negative, zero, or positive, where the transformed critical velocity is $u_{CR} = 1/2a$. The corresponding pressure coefficient is $C_p = -2(\tau/\beta)u$.

The linearized surface boundary condition for the symmetrical parabolic-arc airfoil, whose upper surface is given by $Y_b(x) = \tau F(x) = \tau(0.5 - 2x^2)$ in $-.5 \leqslant x \leqslant .5$ (with $F(x) = 0$ in $|x| > .5$), and the conditions at infinity are

$$v(x,0^+) = F'(x)\,, \tag{4.5a}$$

$$u,v \to 0 \quad \text{as} \quad x^2 + y^2 \to \infty\,. \tag{4.5b}$$

Eq. (4.2a) is written in a "conservation-law" (or divergence) form, in terms of flux components f and g. Therefore discretized forms of (4.2a), for numerical solution, can represent in a fully conservative way either that differential equation or the corresponding integral form. These discretized forms can thus be formulated correctly to represent transitions between elliptic and hyperbolic regions [21,10].

Since only the term $f_x$ in the system (4.2) determines the type of point (depending on the local value of u), one can write the general type-dependent difference equations in the form:

$$(f_x)_T + (g_y)_C = 0 , \quad (u_y)_C - (v_x)_C = 0 , \tag{4.6a,b}$$

where subscript C indicates a central-differenced representation of a derivative and subscript T, which indicates type-dependent differencing, may be replaced by E, H, P, or S at points defined respectively as elliptic, hyperbolic, parabolic, or shock points [21,10]. At all points where the difference equations are clearly elliptic or hyperbolic, subscripts E or H are used. Transition points from elliptic to hyperbolic (progressing downstream from left to right) are P points, and transitions from hyperbolic to elliptic are S points.

For defining the finite-difference operators, Fig. 1 shows a staggered u,v mesh, with the shaded area indicating a mesh cell for eq. (4.2a). The center of a mesh cell is the point at which $\phi$ would be defined on a conventional mesh and is the point that is designated E, H, P, or S. The indices j and k indicate respectively the x and y directions. Second-order-accurate central differences are

$$\left.\begin{aligned} (u_x)_C &= (u_{j,k} - u_{j-1,k})/\Delta x , \quad & (v_y)_C &= (v_{j,k} - v_{j,k-1})/\Delta y , \\ (u_y)_C &= (u_{j,k+1} - u_{j,k})/\Delta y , \quad & (v_x)_C &= (v_{j+1,k} - v_{j,k})/\Delta x . \end{aligned}\right\} \tag{4.7}$$

In general, $(f_x)_T$ is represented by

$$\Delta x (f_x)_T \equiv \Delta f_{j,k} = (f_G)_{j,k} - (f_G)_{j-1,k} \tag{4.8a}$$

where

$$(f_G)_{j,k} = f((u_G)_{j,k}) = (u_G)_{j,k} - a(u_G^2)_{j,k} \tag{4.8b}$$

and where $u_G$ is either a "hyperbolic form" $u_H$ or an "elliptic form" $u_E$. With (i) the definition (4.8) for the difference operator $(f_x)_T$, (ii) a condition to determine whether each $(u_G)_{j,k}$ is represented by $u_E$ or $u_H$, and (iii) specifications of $u_E$ and $u_H$ to obtain the finite differences (4.8a) to the order of

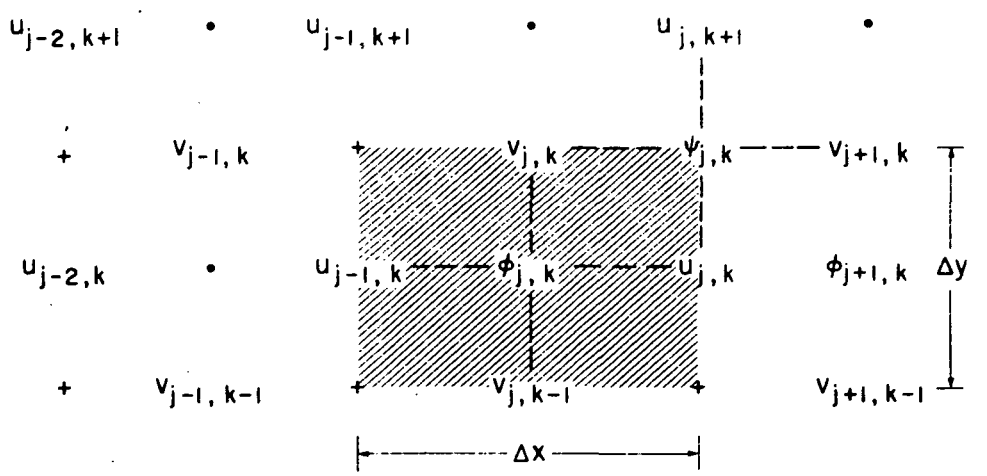


**Fig. 1 – Differencing mesh and mesh cell.**

accuracy desired, all four type-dependent operators are obtained. As derived in [10], (i) the second-order-accurate elliptic operator, (ii) either the first-order or second-order-accurate hyperbolic operator, and (iii) the corresponding parabolic-point and shock-point operators are all produced in (4.6) with (4.7) and (4.8) by the following relationship:

$$(u_G)_{j,k} = (1-\sigma_{j,k})u_{j,k} + \sigma_{j,k}[\lambda u_{j-1,k} + (1-\lambda)u_{j-2,k}] , \tag{4.9}$$

where

$$\left.\begin{aligned} \sigma_{j,k} &= 0 \ (\text{and } u_G = u_E) \quad \text{if} \quad \tilde{u}_{j,k} < u_{CR} , \\ &= 1 \ (\text{and } u_G = u_H) \quad \text{if} \quad \tilde{u}_{j,k} > u_{CR} . \end{aligned}\right\} \tag{4.10}$$

$$\tilde{u}_{j,k} = (u_{j,k} + \delta\, u_{j-1,k})/(1+\delta) . \tag{4.11}$$

and where $\lambda = 1$ for the first-order-accurate hyperbolic operator, $\lambda = 2$ for the second-order-accurate hyperbolic operator, and $\delta$ is a parameter that may be varied from 0 to $\infty$ but is derived as unity for Murman's first-order-accurate operators [21]. As an example to illustrate, suppose $\lambda = 1$, $\delta = 1$, and $\tilde{u}_{j,k} < u_{CR}$ and $\tilde{u}_{j-1,k} > u_{CR}$. Then for the shaded mesh cell in Fig. 1, eqs. (4.8) – (4.10) give

$$\Delta f_{j,k} = u_{j,k} - u_{j-2,k} - a(u^2_{j,k} - u^2_{j-2,k})$$

which is equivalent to Murman's [21] first-order shock-point operator. In a similar way the Krupp-Murman first-order parabolic operator [20] is also obtained. Both the first- and second-order-accurate hyperbolic operators given by (4.8) – (4.10) with $\lambda = 1$ and $\lambda = 2$ are equivalent to upwind difference operators originally proposed by Murman and Cole [19]; the fully conservative second-order P and S operators were introduced in [10]. Analysis of all these E, H, P, and S operators [10] has verified their consistency, accuracy, and stability in the examples computed.

Because of the slow iterative convergence of the second-order-accurate iterative method to be described, two methods of adding artificial viscosity have been proposed and used [10]. Both leave the scheme fully conservative and formally second-order-accurate.

The boundary conditions for the finite-difference equations (4.6) are the same as (4.5) but with (4.5b) replaced by a far-field condition on an outer rectangular boundary B:

$$u_{j,k} = u_B(x,y) \quad \text{or} \quad v_{j,k} = v_B(x,y) \quad \text{on} \quad B \tag{4.12}$$

where, for example, $u_B$ and $v_B$ are given by a Prandtl-Glauert solution (see [15,8,9]).

For solution of eqs. (4.6) with (4.7) through (4.11) and with conditions (4.12) by the semidirect methods, one must rearrange the equations so that the left side is an appropriate elliptic operator and provides a stable iteration scheme. One first adds $(u_x)_C - (f_x)_T$ to both sides of (4.6a) to obtain

$$(u_x)_C + (v_y)_C = (u_x)_C - (f_x)_T , \tag{4.13a}$$

$$(u_y)_C - (v_x)_C = 0 . \tag{4.13b}$$

This set contains a central-differenced elliptic operator on the left side regardless of the local type of the equations. The nonlinear type-dependent term has been shifted to the right side where, in an iterative procedure, it can be computed from a previous iteration. Although the iteration of these equations [8] converged well for subsonic and slightly supercritical flow, it was found [9,10] that terms with parameters multiplying $u_{j,k}$ and $u_{j-1,k}$ needed to be added to both sides of (4.13a) to produce iterative convergence at higher Mach numbers. A more specific form of the difference equations, in which the second-order-accurate relations (4.7) have been substituted, is

$$D_{j,k}(u,v) = R_{j,k}(u), \quad E_{j,k}(u,v) = 0, \tag{4.14a,b}$$

in which

$$D_{j,k}(u,v) \equiv (1-\alpha_1)u_{j,k} - (1+\alpha_2)u_{j-1,k} + \mu^{-1}(v_{j,k} - v_{j,k-1}), \tag{4.15a}$$

$$E_{j,k}(u,v) \equiv (u_{j,k+1} - u_{j,k}) - \mu(v_{j+1,k} - v_{j,k}), \tag{4.15b}$$

$$R_{j,k}(u) \equiv (1-\alpha_1)u_{j,k} - (1+\alpha_2)u_{j-1,k} - \Delta f_{j,k}, \tag{4.15c}$$

and where $\Delta f_{j,k}$ is defined by eqs. (4.8) - (4.11) and $\mu \equiv \Delta y/\Delta x$. The formal order of accuracy of eqs. (4.14) depends on the value of $\lambda$ used in (4.9).

**4.2 Equations for Method 1(a).** As described in section 2.2 above, the straightforward iteration Method 1(a) for eqs. (4.14) is simply

$$D_{j,k}(u_n,v_n) = R_{j,k}(u_{n-1}), \quad E_{j,k}(u_n,v_n) = 0. \tag{4.16a,b}$$

For determining each $\sigma_{j,k}$ in (4.10), eq. (4.11) uses $u_{n-1}$. The presence of $\alpha_1 u_{j,k}$ and $\alpha_2 u_{j-1,k}$ on both sides of eq. (4.16a) allows the interpretation and treatment of these terms as an off-centered time derivative, $\partial u/\partial t$, multiplied by a constant. When the solution converges, these terms cancel out. The semidirect Method 1(a) proceeds by solving the left side of (4.16) in terms of the known right side by an "extended Cauchy-Riemann" solver [18] for $u_n$ and $v_n$ at all points simultaneously. The iteration with $\alpha_1$ or $\alpha_2 \neq 0$ needs a reasonable (but very roughly approximate) initial approximation ($u_o$), such as a Prandtl-Glauert solution. Ref. [10] gives variable specifications of $\alpha_2$ for best convergence.

The boundary conditions on (4.16) are

$$v_n(x,0^+) = F'(x), \tag{4.17a}$$

$$u_n = u_B \quad \text{or} \quad v_n = v_B \quad \text{on} \quad B. \tag{4.17b}$$

**4.3 Equations for Method 2(b).** The artificially extended form, (2.4), of eqs. (4.14) is

$$D_{j,k}(u,v) = (1-\epsilon)R_{j,k}(u_o) + \epsilon R_{j,k}(u), \tag{4.18a}$$

$$E_{j,k}(u,v) = 0. \tag{4.18b}$$

For Method 2(b) assume that

$$u(x,y,\epsilon) = u_1'(x,y) + \epsilon u_2'(x,y) + \epsilon^2 u_3'(x,y) + \dots , \tag{4.19a}$$

$$v(x,y,\epsilon) = v_1'(x,y) + \epsilon v_2'(x,y) + \epsilon^2 v_3'(x,y) + \dots . \tag{4.19b}$$

The successive approximations are then (for $n = 1,2,3 \ldots$)

$$u_n = \sum_{i=1}^{n} \epsilon^{i-1} u_i'(x,y) , \quad v_n = \sum_{i=1}^{n} \epsilon^{i-1} v_i'(x,y) . \tag{4.20}$$

Substitution of (4.19) into (4.18) leads to

$$D_{j,k}(u_n', v_n') = R_{n-1} , \quad E_{j,k}(u_n', v_n') = 0 , \tag{4.21}$$

where:

$$R_o = R_{j,k}(u_o) \tag{4.22a}$$

$$R_1 = R_{j,k}(u_1') - R_{j,k}(u_o) \tag{4.22b}$$

(with $u_o$ being used in (4.11) in determining $\sigma_{j,k}$ for use in $R_{j,k}(u_1')$) and

$$R_2 = (1-\alpha_1)(u_2')_{j,k} - (1+\alpha_2)(u_2')_{j,k} - (\Delta f_2)_{j,k} , \tag{4.22c}$$

$$\Delta f_2 = (f_2)_{j,k} - (f_2)_{j-1,k} . \tag{4.23a}$$

$$\begin{aligned} (f_2)_{j,k} = {} & (1-\sigma_{j,k})\,[(u_2')_{j,k} - 2a(u_1'u_2')_{j,k}] \\ & + \sigma_{j,k}\{[\lambda(u_2')_{j-1,k} + (1-\lambda)(u_2')_{j-2,k}] \\ & - 2a[\lambda(u_1')_{j-1,k} + (1-\lambda)(u_1')_{j-2,k}]\,[\lambda(u_2')_{j-1,k} + (1-\lambda)(u_2')_{j-2,k}]\} . \end{aligned} \tag{4.23b}$$

The boundary conditions are:

$$v_1'(x,0^+) = F'(x) : \quad v_n'(x,0^+) = 0 \ (n = 2,3) : \tag{4.24a}$$

$$u_1' = u_B \quad \text{or} \quad v_1' = v_B \quad \text{on} \quad B ; \tag{4.24b}$$

$$u_n' = 0 \quad \text{or} \quad v_n' = 0 \quad \text{on} \quad B \ (n = 2,3) . \tag{4.24c}$$

With some reasonable approximation for $(u_o)_{j,k}$, such as a nearly converged solution by Method 1(a), eqs. (4.21), with $n = 1,2,3$, give three successive approximations $u_1', u_2', u_3'$ at each j,k to use in (2.12) to obtain an extrapolated solution.

**4.4 Results and discussion.** A research computer program written to solve the transonic small disturbance equations by the methods described above for a biconvex airfoil at zero incidence, includes the option of switching after some iterations by Method 1(a) to the extrapolation technique, Method 2(a). A conversational version of the program, for interacting with the program, was run on an IBM 360/67 computer, and computing times were measured on a Control Data 7600 computer.

Pressure distributions have been computed for a range of subsonic and transonic Mach numbers from both first- and second-order-accurate formulations. Examples by Method 1(a) are shown on Fig. 2 for a thickness ratio of 10 percent and $M_\infty = 0.825$. For this calculation the boundaries were at one-half chord upstream and downstream of the airfoil edges and at 3.5 chords above the airfoil. The results computed on a 39X32 uniform mesh compare well with a line-relaxation program [22], which uses a variable and finer mesh. On a very coarse (19X32) mesh, with only 10 mesh intervals on the airfoil chord, the first-order-accurate results, of course, are not good. The shock is badly smeared, and an anomalous jump behind the sonic point that is characteristic of the first-order P operator is exaggerated on the coarse mesh. However, the second-order-accurate results are very smooth through the sonic point and are surprisingly accurate.

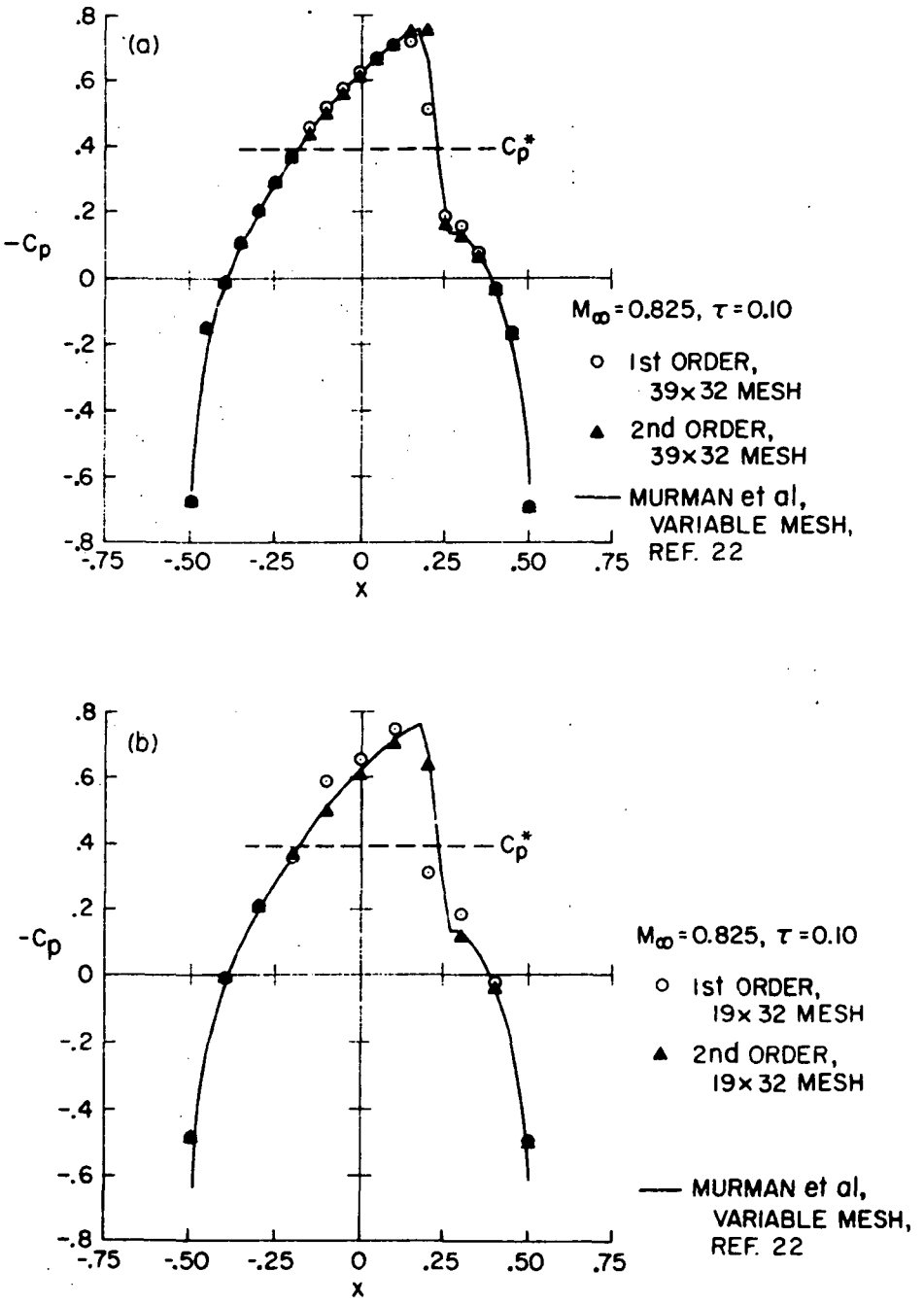


**Fig. 2 – Pressure on a thin biconvex airfoil.**

Figure 3 shows an interesting effect of switching to Method 2 before the iteration has converged enough, when the types of all points are not yet quite the same as the final types. Method 1(a) was used for nine iterations; then Method 2(b) was used to obtain the three successive terms at each point and the extrapolated solution shown in Fig. 3. A property of the Aitken /Shanks extrapolation as used in Method 2 is that all the significant figures of the three successive approximations at any point contain information about the exact solution, even though those successive approximations themselves are not very close to the exact solution (see example problems above in section 3). It thus appears possible in Fig. 3 that this procedure may be picking up the fact that the exact solution to the equations (or the solution on a very fine mesh) has the well-known logarithmic singularity just behind the shock, even though the converged solution on the coarse mesh smears over this singularity. Even the finer mesh used by the program in [22] was not fine enough to pick up the singularity, partly because that point apparently occurs between the mesh points for this case. This phenomenon illustrated in Fig. 3 is not an isolated case but is a typical occurrence in Method 2. It may be that the numerical solution in Fig. 3 is as good as representation of the exact solution to the equations as is the fully converged solution in Fig. 2(a) (circles).

The most significant property of the semidirect method is the relatively short computing time required. On the 39X32 mesh, the time per iteration was measured as 40 milliseconds in a very inefficiently coded program, but for various reasons discussed in [10] it is expected to be reduced to 20 ms. (The direct solver requires only 14 ms) The subcritical cases were sufficiently converged in 3 iterations or less, and a slightly supercritical case (first-order-accurate, using Method 2) required 6 iterations. The

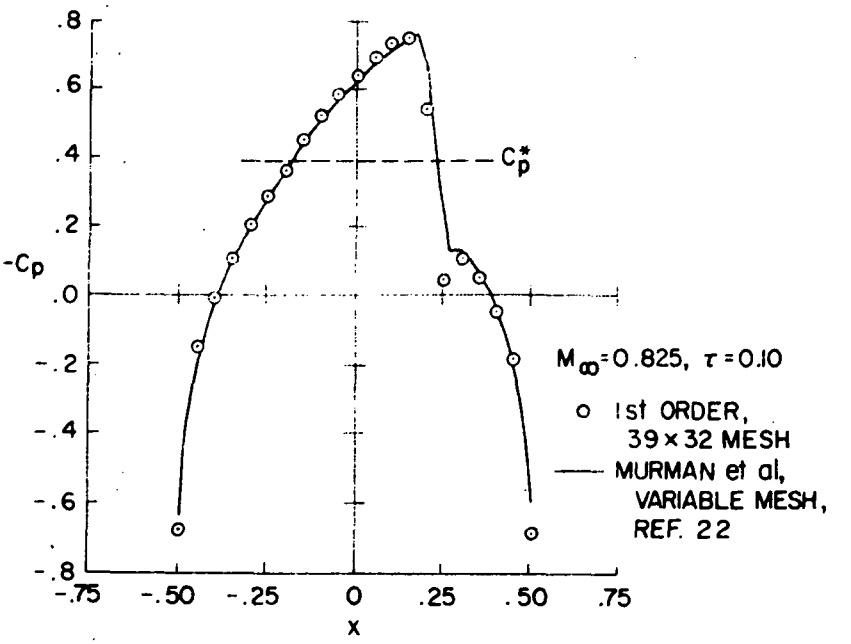


**Fig. 3 – Pressure distribution resulting from Aitken/Shanks extrapolation (Method 2(b)) before iterative convergence.**

first-order-accurate case shown in Fig. 2(a) required 20 iterations by Method 1 and, as described above, the results of Fig. 3 required only 9 iterations by Method 1(a) followed by 3 more by Method 2(b).

At this writing, the program has not yet been written for the above formulation that includes the second-order-accurate formulation in Method 2. It is expected that when this is done, the program can be run rapidly with the first-order ($\lambda = 1$) operators on the very coarse mesh using Method 1, then switched to second-order ($\lambda = 2$) and Method 2 for final extrapolation.

## 5. Concluding Remarks

It has been shown that a special procedure (Method 2) is effective for obtaining most appropriate successive approximations for use in the Aitken extrapolation formula for accelerating the iterative convergence of numerical solutions to nonlinear partial differential equations. The procedure is based on the combined use of artificial perturbation-series expansions and an artificially extended equation. It was shown in a previous paper [8] that one version of the technique was very effective for accelerating iterative convergence when the solutions are smooth. The method, in a modified version, has now been applied with some success to a strongly supercritical transonic flow problem, in which the flow equations are of mixed type and whose solutions have shock-wave discontinuities. The method is expected to be extended to more general flows, including lifting airfoils and three-dimensional flows.

## References


1 Aitken, A.C.: On Bernoulli's numerical solution of algebraic equations. Proc. Royal Soc. Edinburgh 46 (1926) 289–305.

2 Henrici, P.: Elements of Numerical Analysis. John Wiley & Sons, Inc., New York, 1964.

3 Shanks, D.: Nonlinear transformations of divergent and slowly convergent sequences. J. Math. and Phys. 34 (1955) 1–42.

4 Johnson, R. C.: Alternative approach to Padé approximants. Padé Approximants and Their Applications (ed. by P. R. Graves-Morris). Academic Press, London, 1973; 53–67.

5 Genz, A. C.: Applications of the $\epsilon$-algorithm to quadrature problems. Padé Approximants and Their Applications (ed. P. R. Graves-Morris). Academic Press, London, 1973; 105–115.

6 Wynn, P.: On a device for computing the $e_m(S_n)$ transformation. Mathematical Tables and Other Aids to Computation 10 (1956) 91–96.

7 Wynn, P.: Acceleration techniques for iterated vector and matrix problems. Mathematics of Computation 16 (1962) 301–322.

8 Martin, E. D.; Lomax, H.: Rapid finite-difference computation of subsonic and slightly supercritical aerodynamic flows. Presented as a portion of AIAA Paper No. 74-11, 1974. Also AIAA Jour. 13 (1975) 579–586.

9 Martin, E. D.: Progress in application of direct elliptic solvers to transonic flow computations. Aerodynamic Analyses Requiring Advanced Computers, Part II, NASA SP-347, 1975; 839–870.

10 Martin, E. D.: A fast semidirect method for computing transonic aerodynamic flows. AIAA 2nd Computational Fluid Dynamics Conference Proceedings, 1975; 162–174.

11 Bellman, R.: Perturbation Techniques in Mathematics, Physics, and Engineering. Holt, Rinehart and Winston, New York, 1964.

12 Buneman, O.: A compact non-iterative Poisson solver. SUIPR Rept. 294, Inst. for Plasma Research, Stanford Univ., Stanford, Calif., 1969.

13 Hockney, R. W.: The potential calculation and some applications. Methods in Computational Physics, Vol. 9 (ed. by B. Alder, S. Fernbach, and M. Rotenburg). Academic Press, New York, 1970; 135–211.

14 Buzbee, B. L.; Golub, G. H.; Nielson, C. W.: On direct methods for solving Poisson's Equations. SIAM J. Numer. Anal. 7 (1970) 627–656.

15 Lomax, H.; Martin, E. D.: Fast direct numerical solution of the nonhomogeneous Cauchy-Riemann equations. J. Comp. Phys. 15 (1974) 55–80.

16 Hafez, M. M.; Cheng, H. K.: Convergence acceleration and shock fitting for transonic aerodynamics computations. AIAA Paper No. 75-51, 1975.

17 Concus, P.; Golub, G. H.: Use of fast direct methods for the efficient numerical solution of non-separable elliptic equations. SIAM J. Numer. Anal. 10 (1973) 1103–1120.

18 Lomax, H.; Martin, E. D.: Variants and extensions of a fast direct numerical Cauchy-Riemann solver, with illustrative applications. NASA TN D-7934, 1975.

19 Murman, E. M.; Cole, J. D.: Calculation of plane transonic flows. AIAA Jour. 9 (1971) 114–121.

20 Murman, E. M.; Krupp, J. A.: Solution of the transonic potential equation using a mixed finite difference system. Lecture Notes in Physics 8 (ed. by M. Holt). Springer-Verlag, Berlin, 1971; 199–205.

21 Murman, E. M.: Analysis of embedded shock waves calculated by relaxation methods. AIAA Jour. 12 (1974) 626–633.

22 Murman, E. M.; Bailey, F. R.; Johnson, M. H.: TSFOIL-A computer code for 2-D transonic calculations, including wind-tunnel wall effects and wave-drag evaluation. Aerodynamic Analyses Requiring Advanced Computers, Part II, NASA SP-347, 1975; 769–788.